# Steering Angle
# Lane masking and Curvature

ALVITO DANENDRA PUTRA - 2702366395
BEN ARTHUR - 2702377101
NATHAN SHAUN  - 2702363052
WILLIAM HENRY  - 2702377871

## Automotive

# Overview

Lane Based Steering Angle Estimation

**Background :**

- Lane detection is a core component of autonomous driving
- Traditional methods (thresholding, edge detection) are sensitive to:
    - Lighting changes
    - Shadows
    - Complex road textures
- Deep learning enables robust pixel-level understanding of the road

**Project Objectives**

- Design a computer vision system for lane curvater and angle estimation
- Perform lane segmentation using deep learning

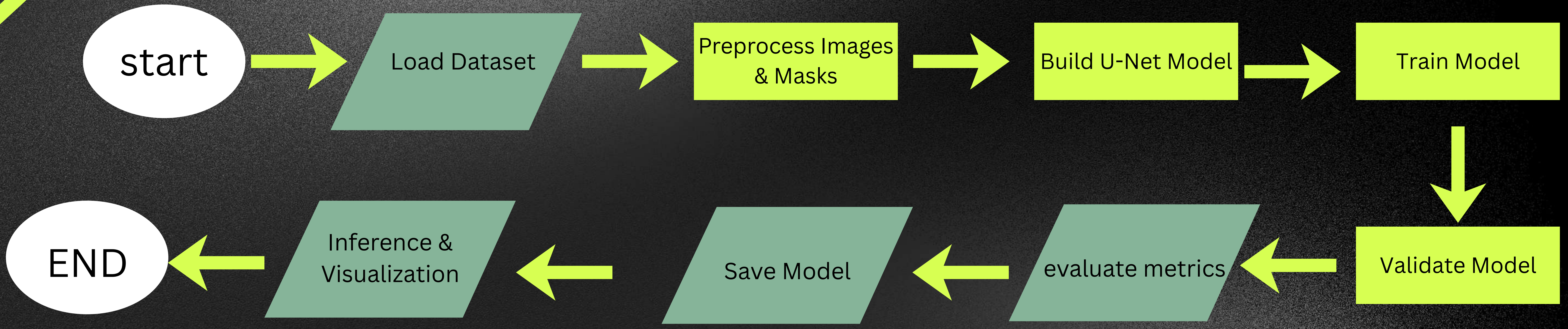
# Overview



**System Overview (pipline) :**

- Camera Image ( stereo vision )
- Preprocessing
- U-Net Segmentation
- Lane Mask
- Boundary Extraction
- Steering Angle Estimation (next stage)

# Flowchart


start
Load Dataset
Preprocess Images & Masks
Build U-Net Model
Train Model
Validate Model
evaluate metrics
Save Model
Inference & Visualization
END

# Datasets

- Dataset  from Roboflow
- Data type: Image–Mask pairs

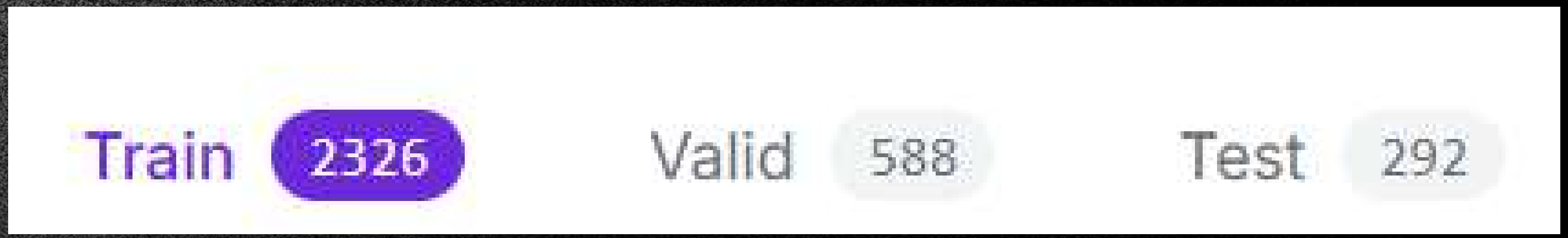
    - RGB road images

    - Binary lane masks

- Dataset split into:

    - Training set

    - Validation set




Characteristics

- RGB road images
- Binary segmentation masks:
    - Lane area (white)
    - Background (black)

Label Definition
White (1): Lane / drivable area
Black (0): Background

# DATA PREPROCESSING

## Image Preprocessing

- Resize images to 256 × 256
- Normalize pixel values to [0, 1]

## Mask Preprocessing

- Convert to grayscale
- Binarization using threshold > 0.5
- Expand dimensions to match model output

```python
# --- image ---
img = Image.open(img_path).convert("RGB")
img = img.resize((img_size, img_size))
img = np.array(img, dtype=np.float32) / 255.0
```

**Preprocessing steps in this project:**

- Encoder extracts lane features
- Decoder reconstructs segmentation mask
- Skip connections preserve spatial details

Output layer:

- 1 channel
- Sigmoid activation

# Model Architecture

**Base model: U-Net (Encoder–Decoder CNN)**

Key components:

- Encoder: extracts hierarchical features
- Bottleneck: compact feature representation
- Decoder: reconstructs spatial details

Skip connections: preserve fine lane boundaries

Feature extraction happen inside the convolution blocks (it learn edges,texture , road geometry )

# Model Architecture

## U-net Modified

Although inspired by the original U-Net, this implementation includes task-specific modifications:

1. Binary Output Channel
   - Output layer uses 1 channel only
   - Designed specifically for binary lane vs background segmentation
2. Sigmoid Activation at Output
   - Produces pixel-wise probabilities in range [0, 1]
   - Enables simple thresholding for lane mask generation
3. Lightweight Architecture
   - Reduced depth and filter size compared to full U-Net
   - Optimized for real-time inference and limited GPU memory
4. Training from Scratch
   - No pretrained weights are used
   - All convolution layers are learned directly from the dataset

# Training (custom u-net)

**Base model: U-Net (Encoder–Decoder CNN)**

**Key components:**

- **Encoder: extracts hierarchical features**
- **Bottleneck: compact feature representation**
- **Decoder: reconstructs spatial details**
- **Skip connections: preserve fine lane boundaries**

- **Training:**
  - **Epochs: 75**
  - **Small batch size (2) due to limited GPU memory**
- **Training Configuration**
- Optimizer: Adam
- Loss Function: Binary Cross-Entropy
- Metrics:
  - Accuracy , Precision , Recall

**DATA PIPELINE**

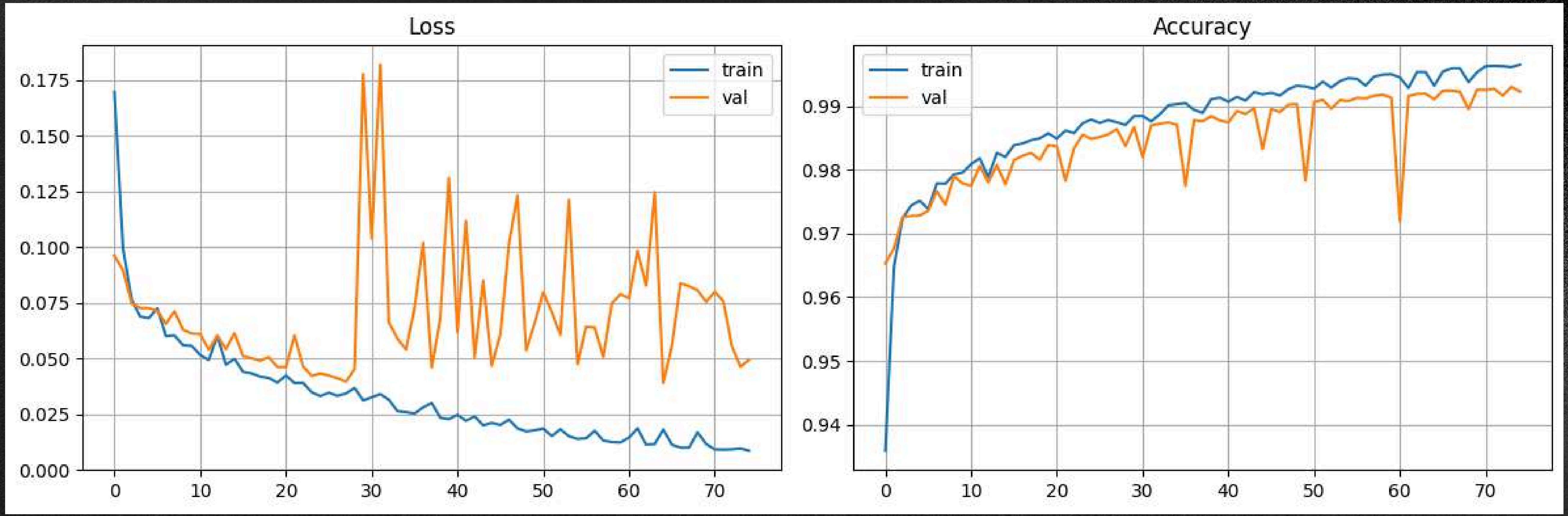
Input Road Image → Image Resizing & Normalization → Convert masks to binary format → Create TensorFlow Dataset → Shuffle, batch, and prefetch data → Feed data into U-Net model for training

# Training Results

## Training Performance

- Training loss decreases steadily
- Validation loss remains stable
- Accuracy reaches above 99%
- No significant overfitting observed

The model shows strong convergence and stability.



# Training Results

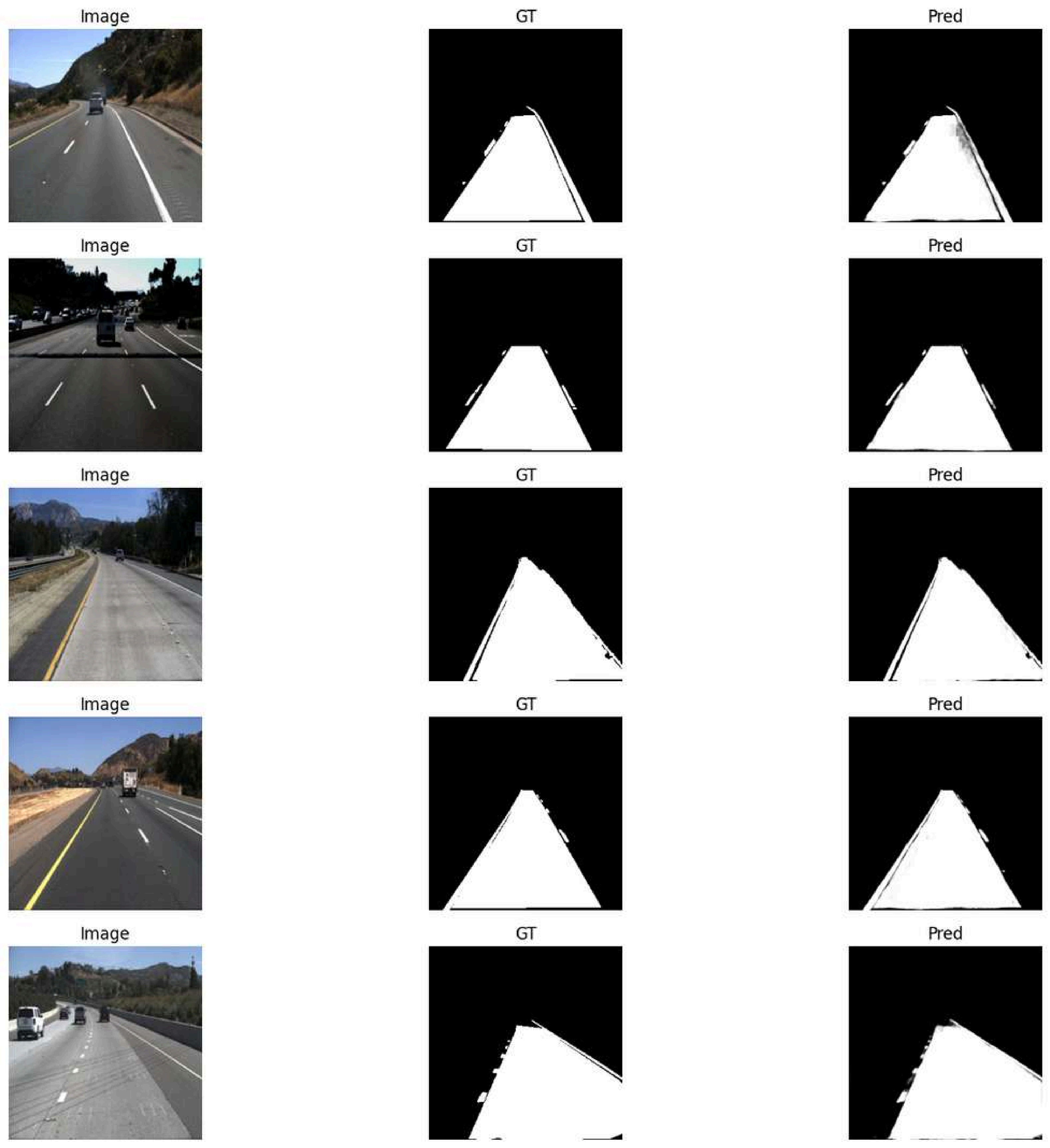
- Predicted lane masks closely match ground truth
- Accurate segmentation across:
  - Straight roads
  - Curved lanes
  - Different lighting conditions

This demonstrates strong generalization ability.

**Model Evaluation**

- High segmentation accuracy
- Good precision and recall
- Consistent validation performance

Evaluation confirms the model is reliable for lane detection.




- **Dataset not from stereo camera**

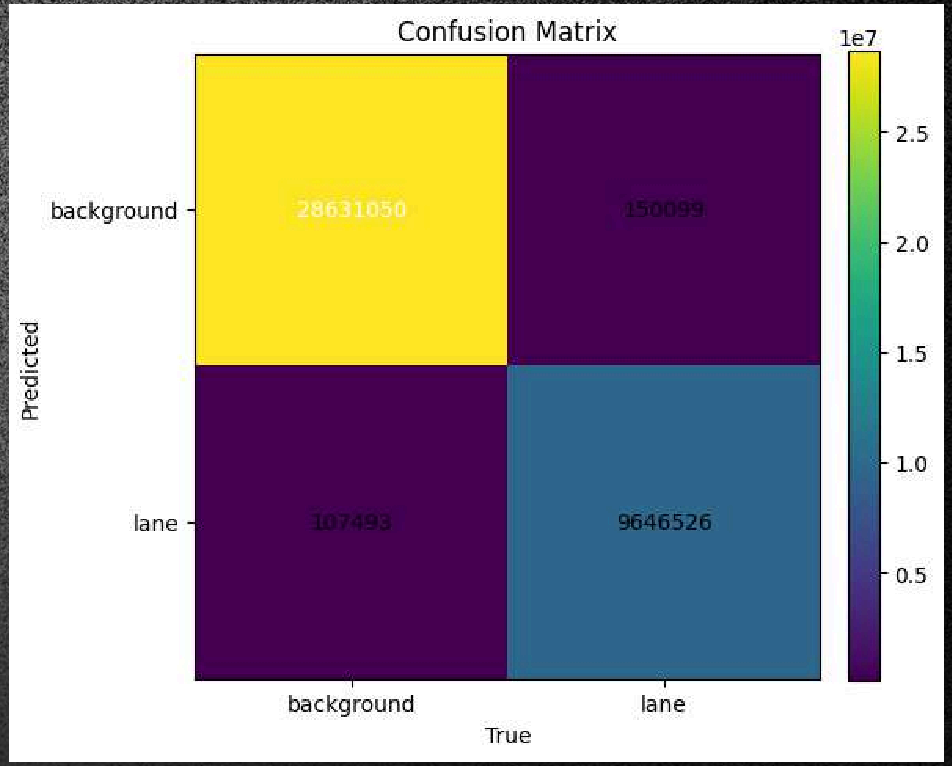
# Training Results

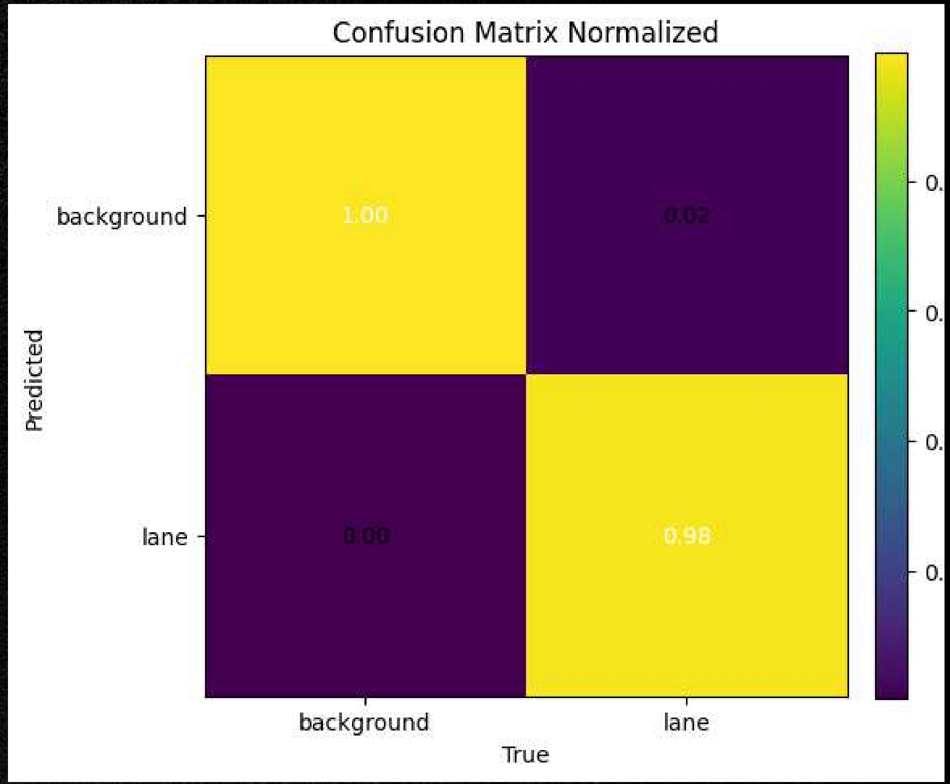
Classes

- Background
- Lane

## Insights

- **High true positive rate for lane pixels**
- **Very low false positives**
- **Balanced classification across classes**




Model Evaluation (Validation Set)

Accuracy : 0.9933
Precision : 0.9847
Recall : 0.9890
F1-Score : 0.9868
IoU (Lane) : 0.9740
Dice (Lane): 0.9868

# conclusions

- Predicted lane masks closely match ground truth
- Accurate segmentation across:
    - Straight roads
    - Curved lanes
    - Different lighting conditions

This demonstrates strong generalization ability.

**Model Evaluation**

- High segmentation accuracy
- Good precision and recall
- Consistent validation performance

Evaluation confirms the model is reliable for lane detection, but **in wet road model is unreliable due to reflection.**

# Sensor Fusion Principles & Components

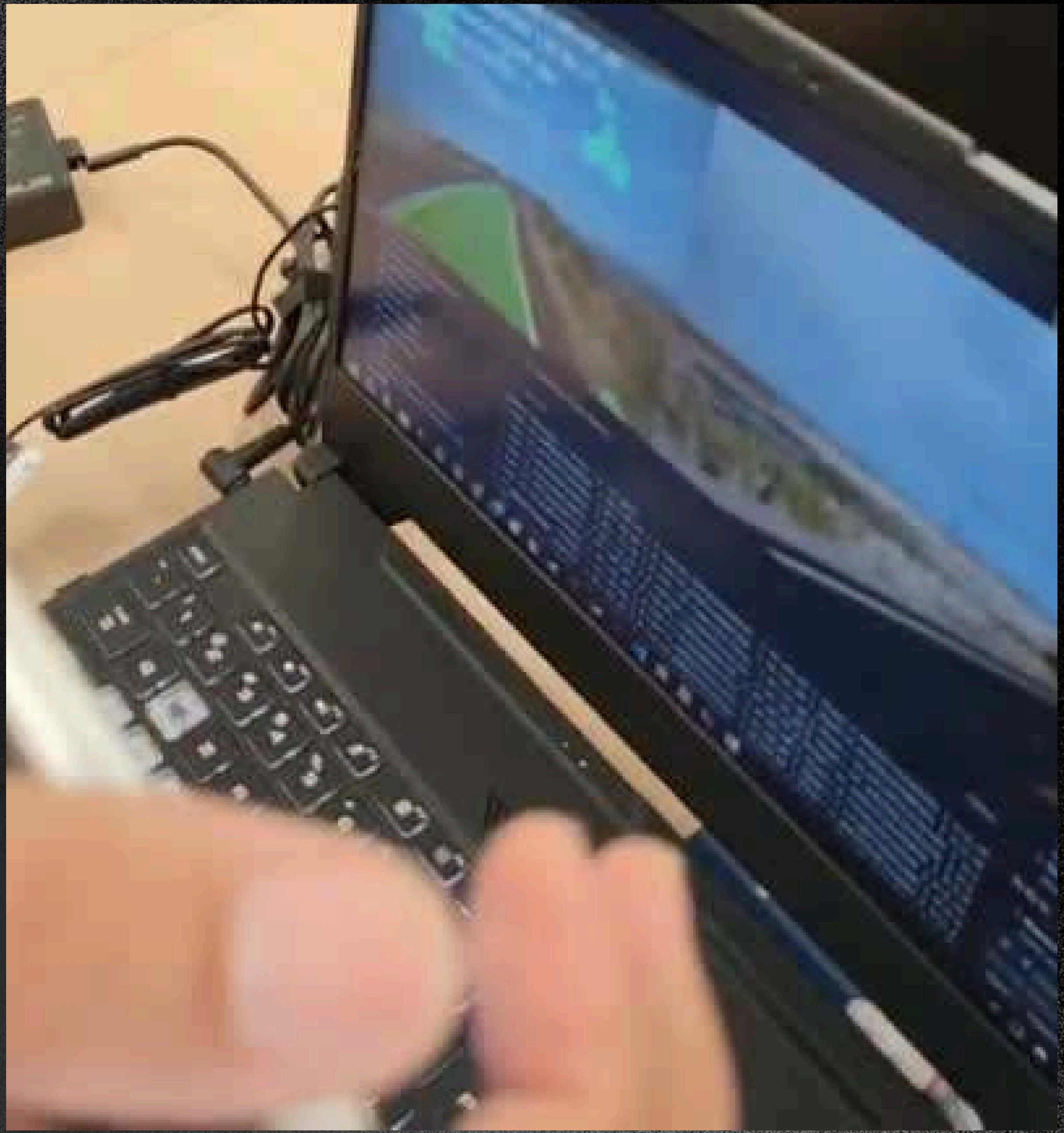
**Limitations**

- No explicit data augmentation (could improve robustness)
- Binary segmentation only (does not separate left/right lane boundaries)
- Performance depends on dataset diversity

**Future work**

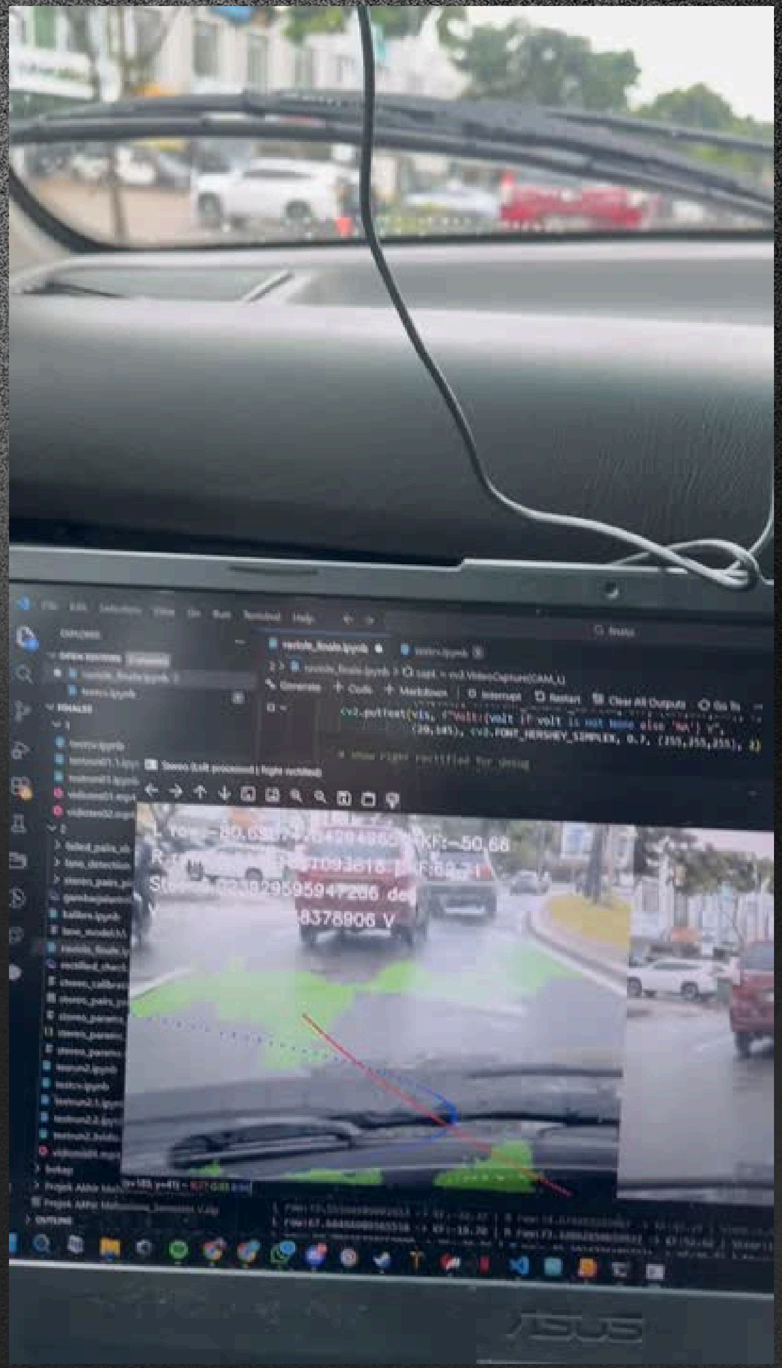
- Add augmentation: brightness/contrast, blur, shadow simulation
- Train multi-class (left lane/right lane/road/background)
- Optimize for real-time inference and deployment

# Results



- Vision-based lane detection combined with Kalman Filter
- Demonstrates practical sensor fusion for vehicle control

# Results



- Vision-based lane detection combined with Kalman Filter
- Demonstrates practical sensor fusion for vehicle control

# Sensor Fusion for Vehicle

# Sensor Fusion Principles & Components

## System Overview

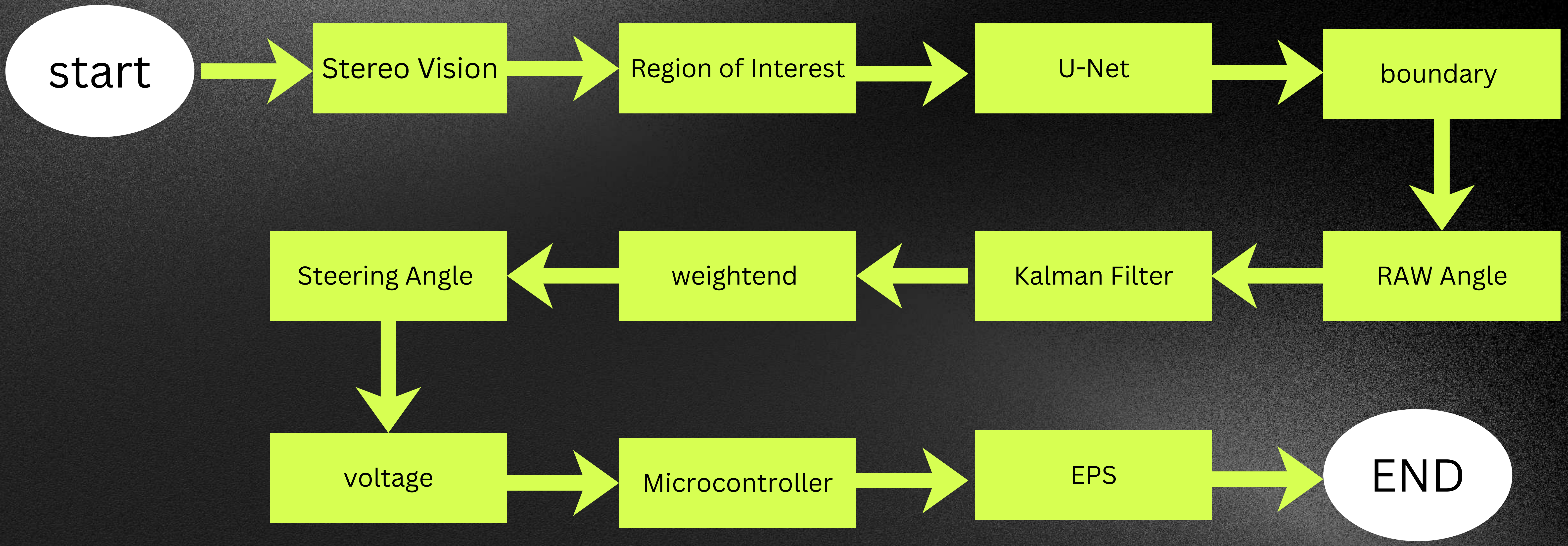
- Sensor: Stereo Vision Camera (2 webcams)
- Perception Model: Semantic Segmentation (U-Net)
- Feature Extraction: Lane boundary detection
- Estimation Layer: Kalman Filter
- Control Output: Steering Angle & Voltage

**Objective:**

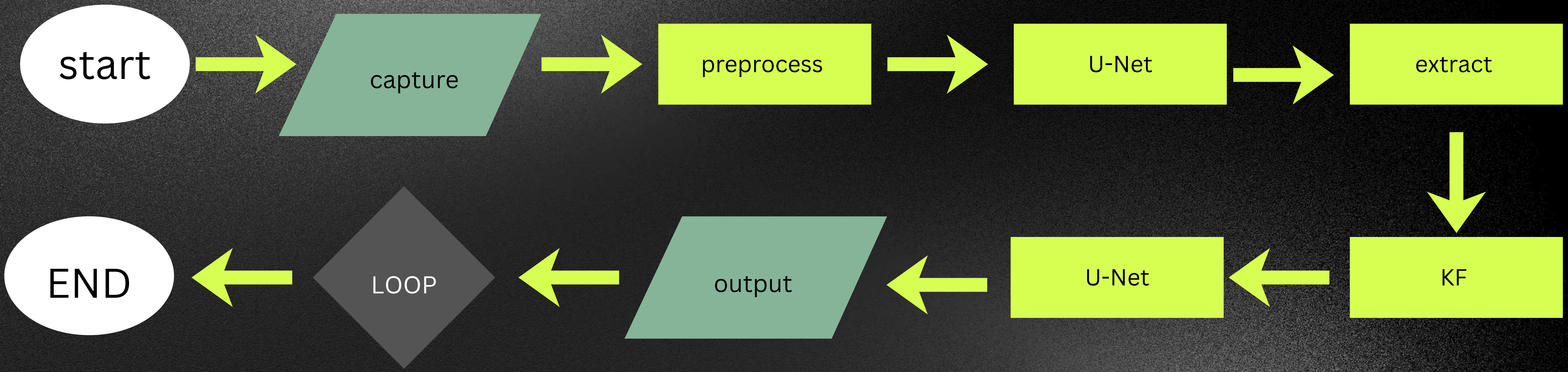
- Stereo camera → lane perception
- Kalman Filter → noise reduction
- Angle estimator →  output

Stereo Camera → U-Net Segmentation → Lane Boundary → Angle Measurement → Kalman Filter → Fusion → θ_steer → Command (MCU/EPS)

# Block Diagram



# Flowchart


start
capture
preprocess
U-Net
extract
KF
U-Net
output
LOOP
END

# Linear & Nonlinear Models

## Nonlinear Models

- Convolutional Neural Network (U-Net)  (Mengubah gambar jalan menjadi binary lane mask)
- Polynomial curve fitting (2nd order)
- Angle computation

## Linear Models

- Kalman Filter (state-space linear model)
- Steering angle to voltage mapping (linear scaling)

# State-Space Modeling

## State Equation

The steering angle estimation is modeled as:

$$x_k = x_{k-1} + w_k$$

## Measurement Equation:

$$z_k = x_k + v_k$$

xk = sudut sebenarnya pada frame camera ke-k
xk−1 = sudut pada frame camera sebelumnya
wk = noise proses (perubahan kecil antar frame camera)
zk = hasil pengukuran sensor (stereo camera)
vk = noise sensor

# Kalman Filter Development

2D Discrete Kalman Filter

## Prediction

Input Kalman Filter:
zL(k): sudut boundary kiri
zR(K): sudut boundary kanan

## Update

### Steering Angle Estimation:

- Adjustable weights allow confidence tuning
- Output is converted to steering voltage

Data Source:

- Frame camera → U-Net → lane mask (frame masuk ke u-net)
- Mask → boundary extraction (mengunah area lane menjadi garis tepi)
- Boundary → curve fitting (menghaluskan boundery)
- Curve → tangent angle (menentukan arah lajur relatif terhadap kendaraan)

Final Output:

- Steering Angle (degree)
- Steering Voltage (V)

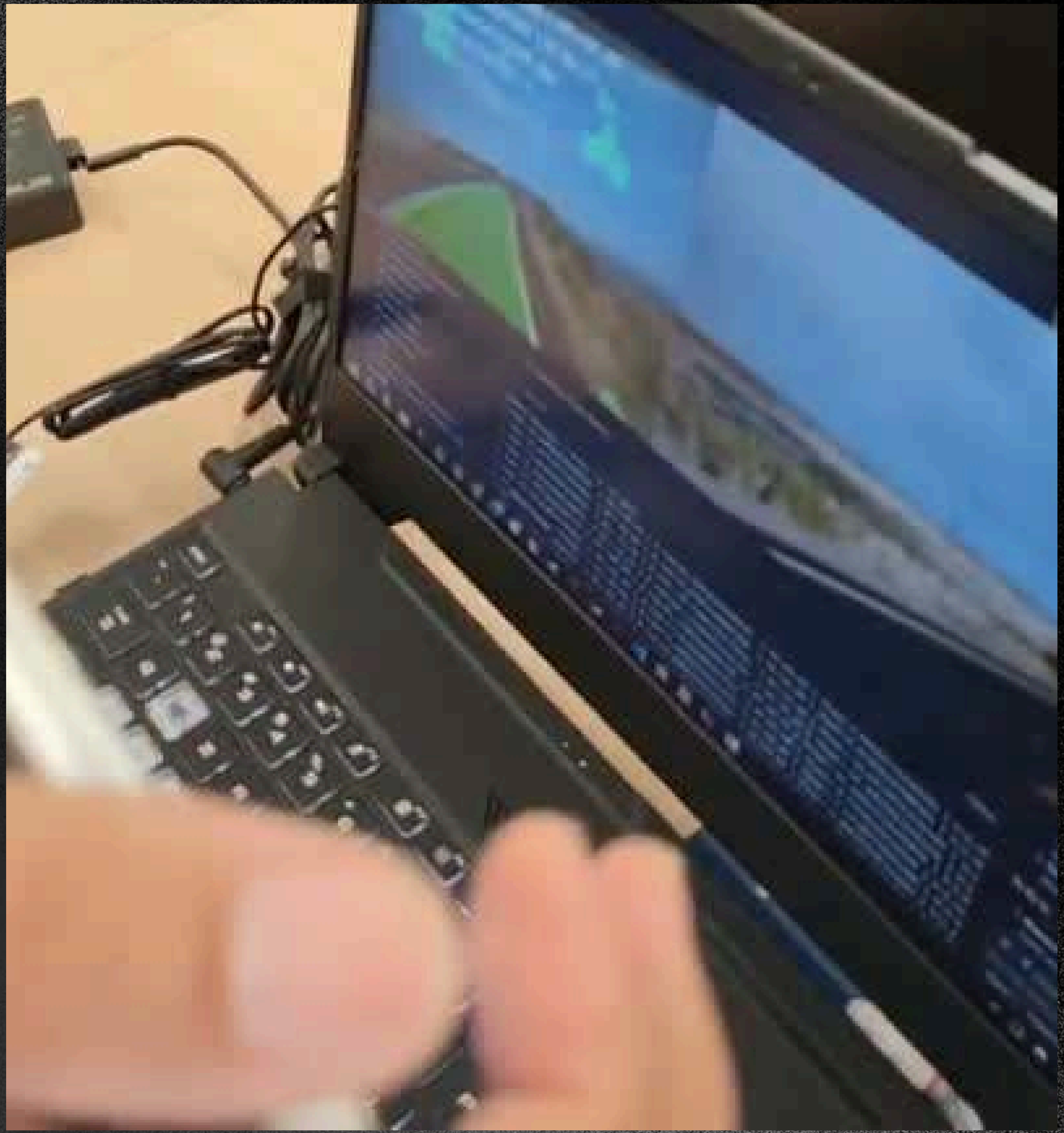
# Validation & Results

- Masking visualizes drivable lane area
- Curved lane boundaries extracted
- Kalman Filter smooths angle fluctuation
- Steering output stable even on curved roads
- Observed Benefit:
- Reduced noise
- More realistic steering respons

# Conclusion

- Vision-based lane detection combined with Kalman Filter
- Demonstrates practical sensor fusion for vehicle control

# Results

- Vision-based lane detection combined with Kalman Filter
- Demonstrates practical sensor fusion for vehicle control


Autonomous Vehicle

# System Overview

Latar Belakang:

- Steering merupakan komponen utama lateral control pada autonomous vehicle
- Estimasi sudut kemudi

Tujuan

- Menghasilkan Estimated Steering Angle
- Berbasis stereo vision secara independen

Overview Sistem

Stereo Camera → Perception → Estimation → Control → Actuation (eps)

# Project Component

**Stereo Camera**

- Dua kamera (Left & Right)
- Digunakan untuk: Lane detection , Heading and curvature estimation

**Microcontroller** : esp32

Sensing

- Stereo Camera (Left–Right)
- Input visual lingkungan jalan

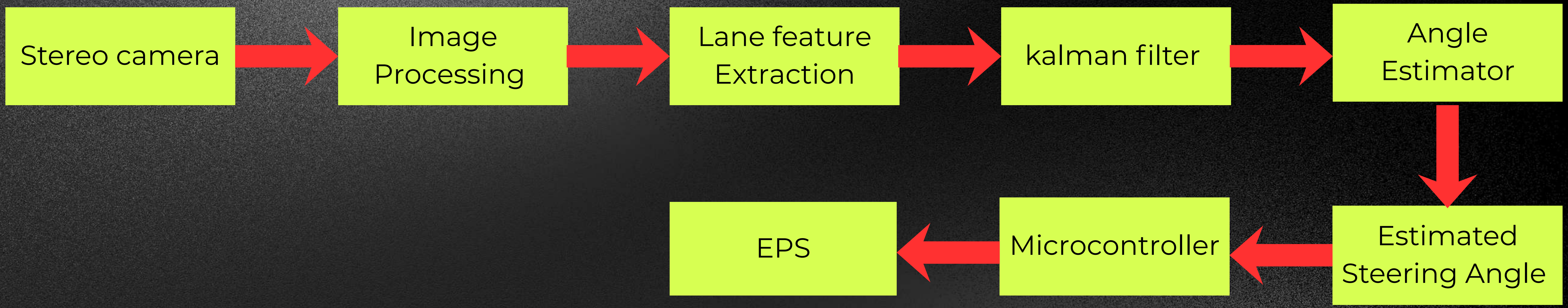
Perception

- Image preprocessing
- ROI (Region of Interest) pada area jalan
- Road & lane masking

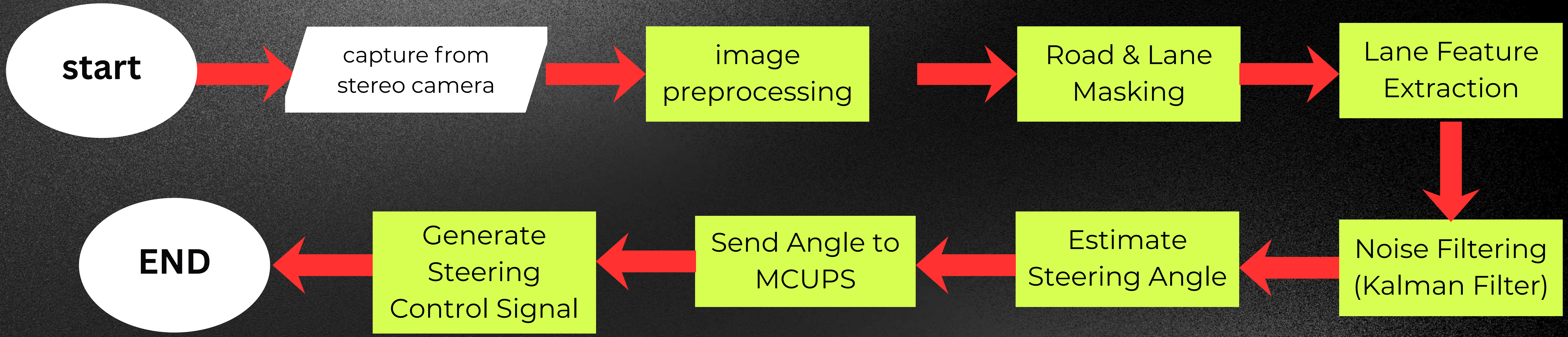
Control & Output

- Estimated Steering Angle :
  - MCU (Microcontroller Unit)
  - contoh: ESP32
- MCU menghasilkan:
  - PWM (Pulse Width Modulation)
- Digunakan untuk aktuator steering (nema23)

# Block Diagram



# Flowchart


start
capture from stereo camera
image preprocessing
Road & Lane Masking
Lane Feature Extraction
Noise Filtering (Kalman Filter)
Estimate Steering Angle
Send Angle to MCUPS
Generate Steering Control Signal
END

# Journey

- Unable to give input signals to eps and eps controller
- using esp as microcontroller and to give input for the stepper
- tb660 as the stepper motor driver
- next improvement make the computer vision , sensor fusion and autonomous steering to be synchronize

**Model Evaluation**

- Good accuracy and masking for lane segmentation and angle curvature
- Adaptor material for stepper should be improve
- Evaluation confirms the model is reliable for lane detection, but **in wet road model is unreliable due to reflection.**

# Validation & Results

- Masking visualizes drivable lane area
- Curved lane boundaries extracted
- Kalman Filter smooths angle fluctuation
- Steering output stable even on curved roads
- Observed Benefit:
- Reduced noise
- More realistic steering respons

# Conclusion

- Vision-based lane detection combined with Kalman Filter
- Demonstrates practical sensor fusion for vehicle control

# Validation & Results

# Thank YOU